CS7026: Authoring for Digital Media

Markup Languages
Yesterday we looked at developing a markup language for a poetry anthology using SGML.

If a DTD describes a sufficiently large class of documents it may make sense to construct display software specifically for that class of document and to hardwire the layout rules into it.

That way every document will be laid out consistently by a particular display program.
This was the approach followed by the first generation of web browsers.

Most of the formatting of headers, lists, tables and other elements was controlled by the browser, but some details, specifically fonts, type sizes and colours, could be controlled by preferences set by the user.
Alternative option: Provide a separate specification of the layout, parallel to and complementing the DTD.

This layout specification is known as a stylesheet.

For each tag defined in the DTD, a stylesheet provides a rule describing the way in which elements with that tag should be laid out.
Since structural markup can be interpreted on any platform in a logically consistent – if not necessarily visually consistent – manner, it is ideally suited for the markup of documents which are to be transmitted over networks.

When the World Wide Web was being designed SGML was used as the basis of the Hypertext Markup Language (HTML).
HTML

- HTML provides a set of tags suitable for marking up web pages.

- It does not provide any way of defining new tags.

- HTML is defined by an SGML DTD, which describes the structure of the class of documents known as Web pages.

- Like SGML it is concerned solely with identifying the structural elements of a page and not with their appearance.
Originally the WWW was intended as a means of disseminating scientific research, so the DTD for the Web pages contained tags corresponding to the main elements of a scientific paper:

- Title
- Headings that can be nested several levels deep
- Lists of various types
- Some typographical control (bold, italic)
- Tables
- Forms
- Etc.
HTML’s Limitations

- Although adequate for marking up simple mainly textual papers for which they were originally intended, HTML’s layout tags are not sufficient for all the diverse types of material which have found their way onto the WWW.

- Two results:
  - Browser manufacturers added *ad hoc* proprietary extensions to HTML.
  - Web Designers used tags as they were never intended.
XML

- Proposed Solution – provide web designers with a facility to define their own tags.

- SGML can do this, but too unwieldy, slow response times.

- Work on adapting SGML to the Internet led to the definition of a subset, known as XML (eXtensible Markup Language).
XML

- Provides all of the important facilities of SGML without the overhead imposed by full SGML.

- XML allows developers to define their own DTDs for any type of document.

- All new browsers now read and interpret (parse) both HTML and XML.
XHTML

- XHTML is an XML-based language that reproduces essentially all the features of HTML, subject to the stricter syntax of XML.

- XHTML documents can be processed by both HTML and XML software. So XHTML documents can be read by existing HTML browsers or can be used with software that understands XML.
XHMTL

The Most Important Differences:
- XHTML elements must be properly nested
- XHTML elements must always be closed
- XHTML elements must be in lowercase
- XHTML documents must have one root element
SGML and XML Family Tree
And Then There Was...

- **HTML5** was the next major revision of the HTML standard.

- Like its immediate predecessors, HTML 4.01 and XHTML 1.1, HTML5 is a standard for structuring and presenting content on the World Wide Web.
Whose Child is it Anyway?

- HTML 4.0 is based on the SGML Rulebook
- XHTML is based on the stricter XML Rulebook

- HTML5 defines a markup language that isn’t based on either rulebook.

- It can be written in either HTML form (or “serialization”) or XHTML form.
How Did HTML5 Get Started?

- HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG) a loose and open collaboration of Web browser manufacturers and interested parties.

- WHATWG was working with web forms and applications, and W3C was working with XHTML 2.0. In 2006, they decided to cooperate and create a new version of HTML.
How Did HTML5 Get Started?

- Some rules for HTML5 were established:
  - New features should be based on HTML, CSS, DOM, and JavaScript
  - Reduce the need for external plugins (like Flash)
  - Better error handling
  - More markup to replace scripting
  - HTML5 should be device independent
  - The development process should be visible to the public

- HTML5 reached recommendation status on 28th October 2014
What is HTML?

- Can be applied to text, images, sound, and video files - in fact to almost any kind of electronic information.

- Used to structure documents and link them together regardless of the type of computer with which the file was originally created.

- Uses predefined commands to determine how pieces of the document should be structured - known as elements or tags.

- W3C - The World Wide Web Consortium: [http://www.w3.org/](http://www.w3.org/)
How Does it Work?

- Separates text from the structure of the text.

- Web browser interprets the structure after you receive the page - information comes to you faster.
How Does it Work?

Steps

1. Using your web browser you go looking for information.
2. Browser finds computer that stores this information (server) by using the HTTP protocol.
3. Server sends the Web page as a plain text file back to you (again using HTTP).
4. Browser interprets the text and tags and shows you the formatting, graphics and text that appears on the page.
Creating your First Page

- Open Notepad++:

```html
<html>
  <head>
    <title>My first html document</title>
  </head>
  <body>
    <p>This is my first html document</p>
  </body>
</html>
```
Creating your First Page (cont.)

- The `<html>` and `</html>` tags identify your file as a HTML file.

- The `<head>` and `</head>` tags are used to indicate any information about the document itself (more anon...).

- The `<title>` and `</title>` tags are used to add a title to your browser’s title bar (The coloured band at the top of your browser).
Creating your First Page (cont.)

- The `<body>` and `</body>` tags are used to surround any content that will appear within the HTML page.

- The `<p>` and `</p>` tags indicate that the enclosed text is a paragraph.
If you were to create another simple web page, you would see that the same HTML tags are present in this document. Only the text that appears between each pair of tags is changed:

```html
<html>
<head>
<title>My Second Web Page</title>
</head>
<body>
<p>This is my second Web page</p>
</body>
</html>
```
Required Tags

- All of these tags are required except the `<p></p>` tags.

- Every XHTML page **must** include the following pairs of tags:
  - `<html>  </html>`
  - `<head>  </head>`
  - `<title>  </title>`
  - `<body>  </body>`
In order to view your page in a browser you must first save it.

All HTML documents must be saved using a `.htm` or `.html` extension.

You can preview any HTML file in your browser, even when that file is stored on your computer rather than the web server.
Saving and Viewing Your Document

- Firefox - select ‘open file’ from the File menu.

- Internet Explorer - select ‘open’ from the File menu, then ‘browse’ for your file.

- Google Chrome – use ctrl + o to launch the Open Window
How Browsers Work

- Although all HTML commands are the same, not all browsers interpret the commands in the same way.

- Some browsers do not understand all the newer HTML 5 Commands - may produce errors.

- However most Web pages will look the same regardless of the browser or platform - just test, test, test…

  [http://validator.w3.org/](http://validator.w3.org/)
Remember the Basics

- HTML is very forgiving… this is great for beginners but if you want to progress you should follow some basic Web coding principles (more to follow…)
  - Include all the required HTML elements that you learned in this lecture.
  - Use lowercase for all filenames (for your convenience).
  - Never use spaces in filenames.
How do you add another paragraph?

Web browsers don’t pay any attention to how many blank lines you put in your text; they only pay attention to the HTML commands.

The `<p></p>` tag tells the browser to add a blank line before it displays any text that follows.
Adding Text and Formatting

<html>
<head>
<title>Typing Paragraphs in HTML</title>
</head>

<body>
<p>This is the first line. But is this the second?</p><p>No, this is.</p>
</body>
</html>
Adding Text and Formatting

This is the first line. But is this the second?

No, this is.
Adding Text

- What if you just want your text to appear on the next line? (lines of an address, poetry etc)…

- Use the line break, or `<br>` tag - forces the browser to move any text following the tag to the next line of the browser, without adding a blank line in between.

- However…
Closing Tags

- *Every open tag in your XHTML document needs to be accompanied by a closing tag.*
- So, a paragraph in your document cannot look like this:
  
  `<p>This is a bad paragraph!</p>`
- Instead, use a closing tag, like this:
  
  `<p>This is a good paragraph!</p>`
Closing Tags

- There is a shortcut for closing tags that don't usually need to be logically closed. The shortcut is to add a close-command slash at the end of the tag to turn it off.

- For instance, there's no need to type: `<br></br>` In XHTML, the tag looks like this: `<br />`
This shortcut has been accepted as the de facto standard when using tags that don't normally appear with closing tags when used in HTML.

While something like a `<br></br>` set of tags is valid in the XML world, they can often confuse browsers that were written without XML in mind. So, just use `<br />` to be safe (don't forget the space before the `/` character).

This rule applies to all tags that you wouldn't normally close if you were writing in HTML 4.
Adding Text and Formatting

- **Headings**
  - HTML enables you to add six different heading tags to your pages by using the tags `<h1>` through to `<h6>`. These tags have an associated closing tag:

    `<h1>This is Heading 1</h1>`

  - The closing tags also create an automatic paragraph break; in other words all headings automatically include a blank line to separate them from the next text.

  - By default, Heading 1, the `<h1>` tag is the largest font of the heading tags and Heading 6, the `<h6>` tag the smallest.
Adding Text and Formatting

Heading 1

Heading 2

Heading 3

Normal Text

Heading 4

Heading 5

Heading 6
Exercise

- Create the XHTML page pictured on the following slide.
- Include all the required HTML elements that you learned in this lecture.
- Save it and view it in your browser.
Beautiful

Beautiful

is the
unmeaning
of(sil
ently)fat

ling(c
ver
yw
here)s

Now

by ee cummings